CLAIMS

What is claimed is:

1   1.   A method for dynamically extending element types for a voice-based

2   extensible mark-up language (VoiceXML), comprising:

3   (a)   registering a plurality of element types with a VoiceXML interpreter;

4   (b)   receiving the element types during use of the VoiceXML interpreter; and

5   (c)   accessing code associated with the registered element types utilizing the

6   VoiceXML interpreter;

7   (d)   wherein the code extends the functionality of the VoiceXML.

1   2.   The method as set forth in claim 1, wherein the code is written in JAVA.

1   3.   The method as recited in claim 1, wherein the registration includes tagging

2   the registered element types as being extensions to a conventional set of

3   element types.

1   4.   The method as recited in claim 3, wherein the element types are tagged

2   utilizing extensible mark-up language (XML) namespaces.

1   5.   The method as recited in claim 3, wherein the registration includes

2   identifying a VoiceXML element type to be extended.

1   6.   The method as recited in claim 5, wherein the registration includes

2   identifying a name for the VoiceXML element type to be extended.

1   7.   The method as set forth in claim 6, wherein the registration includes

2   identifying a class to be loaded for the VoiceXML element type to be

3   extended.

1   8.   The method as set forth in claim 6, wherein the registration includes

2   identifying a location of a file containing class files associated with the

3   identified class.

1   9.   The method as set forth in claim 1, wherein the VoiceXML interpreter is a

2   component of a speech recognition/synthesis system available over the

3   Internet.

1   10.   A computer program product for dynamically extending element types for a

2   voice-based extensible mark-up language (VoiceXML), comprising:

3   (a)   computer code for registering a plurality of element types with a VoiceXML

4   interpreter;

5   (b)   computer code for receiving the element types during use of the VoiceXML

6   interpreter; and

7   (c)   computer code for accessing code associated with the registered element

8   types utilizing the VoiceXML interpreter;

9   (d)   wherein the code extends the functionality of the VoiceXML.

1   11.   The computer program product as set forth in claim 10, wherein the code is

2   written in JAVA.

1   12.   The computer program product as recited in claim 10, wherein the

2   registration includes tagging the registered element types as being extensions

3   to a conventional set of element types.

1   13.   The computer program product as recited in claim 13, wherein the element

2   types are tagged utilizing extensible mark-up language (XML) namespaces.

1   14.   The computer program product as recited in claim 13, wherein the

2   registration includes identifying a VoiceXML element type to be extended.

1   15.   The computer program product as recited in claim 14, wherein the
2   registration includes identifying a name for the VoiceXML element type to
3   be extended.

1   16.   The computer program product as set forth in claim 15, wherein the
2   registration includes identifying a class to be loaded for the VoiceXML
3   element type to be extended.

1   17.   The computer program product as set forth in claim 15, wherein the
2   registration includes identifying a location of a file containing class files
3   associated with the identified class.

1   18.   The computer program product as set forth in claim 10, wherein the
2   VoiceXML interpreter is a component of a speech recognition/synthesis
3   system available over the Internet.

1   19.   A system for dynamically extending element types for a voice-based
2   extensible mark-up language (VoiceXML), comprising:
3   (a)   logic for registering a plurality of element types with a VoiceXML
4   interpreter;
5   (b)   logic for receiving the element types during use of the VoiceXML
6   interpreter; and
7   (c)   logic for accessing code associated with the registered element types utilizing
8   the VoiceXML interpreter;
9   (d)   wherein the code extends the functionality of the VoiceXML.

1   20.   A method for dynamically extending element types for a voice-based
2   extensible mark-up language (VoiceXML), comprising:
3   (a)   registering a plurality of element types with a VoiceXML interpreter utilizing
4   a data structure including:

5 (i) a VoiceXML element type to be extended,

6 (ii) a name for the VoiceXML element type to be extended,

7 (iii) a class to be loaded for the VoiceXML element type to be extended,

8 and

9 (iv) a location of a file containing class files associated with the identified

10 class;

11 (b) tagging the registered element types as being extensions to a conventional set

12 of element types, wherein the element types are tagged utilizing extensible

13 mark-up language (XML) namespaces;

14 (c) receiving element types during use of the VoiceXML interpreter;

15 (d) determining whether the received element types are registered based on the

16 tagging; and

17 (e) accessing code associated with the element types utilizing the VoiceXML

18 interpreter if the received element types are determined to be registered;

19 (f) wherein the code extends the functionality of the VoiceXML.

1 21. A data structure stored in memory for dynamically extending element types

2 for a voice-based extensible mark-up language (VoiceXML), comprising:

3 (a) a VoiceXML element type object for identifying a VoiceXML element type

4 to be extended;

5 (b) a name object for identifying a name for the VoiceXML element type to be

6 extended;

7 (c) a class object for identifying a class to be loaded for the VoiceXML element

8 type to be extended; and

9 (d) a location object for identifying a location of a file containing class files

10 associated with the identified class;

11 (e) wherein the data structure is capable of being used to register element types

12 capable of accessing code to extend the functionality of the VoiceXML.

1 22. A method for dynamically extending a type attribute of elements of a voice-

2 based extensible mark-up language (VoiceXML), comprising:

3   (a)   registering with a VoiceXML interpreter an extended type attribute

4   associated with an element of VoiceXML;

5   (b)   receiving the element during use of the VoiceXML interpreter;

6   (c)   identifying the extended type attribute associated with the element; and

7   (d)   accessing code corresponding to the registered type attribute utilizing the

8   VoiceXML interpreter;

9   (e)   wherein the code extends the functionality of the VoiceXML.

1   23.   A computer program product for dynamically extending a type attribute of

2   elements of a voice-based extensible mark-up language (VoiceXML),

3   comprising:

4   (a)   computer code for registering with a VoiceXML interpreter an extended type

5   attribute associated with an element of VoiceXML;

6   (b)   computer code for receiving the element during use of the VoiceXML

7   interpreter;

8   (c)   computer code for identifying the extended type attribute associated with the

9   element; and

10   (c)   computer code for accessing code corresponding to the registered type

11   attribute utilizing the VoiceXML interpreter;

12   (d)   wherein the code extends the functionality of the VoiceXML.

1   24.   A data structure stored in memory for dynamically extending a type attribute

2   of elements of a voice-based extensible mark-up language (VoiceXML),

3   comprising:

4   (a)   a VoiceXML type attribute object that is extended to include a previously

5   undefined type attribute;

6   (b)   a VoiceXML element; and

7   (c)   a class object for identifying a class to be loaded for the VoiceXML type

8   attribute object that is extended;

9   (d)   wherein the data structure is capable of being used to register VoiceXML

10   type attribute objects capable of accessing code to extend the functionality of

11   the VoiceXML.

1   25.   The data structure as set forth in claim 24, wherein the element includes at

2   least one of grammar and field.

1   26.   The data structure as set forth in claim 24, wherein the type includes at least

2   one of digits, number, phone, currency, equity, airline information, address,

3   and country.